moving rp_filter into netfilter

xt_rpfilter match

Florian Westphal

August 23, 2011

```
   1024D/F260502D <fw@strlen.de>
1C81 1AD5 EA8F 3047 7555
E8EE 5E2F DA6C F260 502D
```

Abstract

RFC3704 describes a method for Ingress Filtering for Multihomed Networks using Reverse Path Forwarding. The Linux kernel implements this in the ipv4 input routing path, but it has been suggested that it should be moved into the firewalling code. As usual, pesky details are getting in the way.

Outline

- Problem Statement
- Current Implementation
- `xt_rpfilter`, `ipt_rpfilter` attempts
- Remaining Issues / Open Questions

Problem Statement

- In kernel `rp_filter` only supports ipv4
- patches that add ipv6 aquivalents were rejected. davem:
  - "when I remove the routing cache from ipv4, I want to get rid of RP filtering entirely"
  - "I think the BSD guys did the right thing, and put this in the firewalling code"
- → `rpfilter` match for netfilter

Current Implementation

- implemented in `fib_validate_source`, called via `ip_route_input` path
- also handles local/bcast/mcast packets
- takes advantage of rt cache (cache entry exists = already passed rpfilter)
- → no more rt-cache = additional per-packet `fib_lookup`
- controlled via sysctl (`conf.$dev.rp_filter`, `..src_valid_mark`, `..accept_local`)
- supports a strict (best reverse path via iif) and loose (any interface) mode

Problems with netfilter rpf module, Part 1

Wanted: pure reverse lookup; given src s, dst d, iif dev:
Would source d, destination s, pass via oif dev?

- can't use iif in `ip_route_output_key`, setting oif doesn't work either (interface route!)
- can't use `ip_route_input`: skb is `const`, oif unknown in `PRE_ROUTING`
- can't restrict to `FORWARD`: forward path can send icmp errors before nf hook invocation (ttl, PMTUD, . . .)
- successful lookup via `ip_route_xxx` creates rt cache entry (not desireable)

1st Try

nothing fancy: pure route lookup, compare selected interface

```
flowi4_init_output(&flow, 0, mark, RT_TOS(iph->tos),
                   0, iph->protocol, FLOWI_FLAG_ANYSRC,
                   iph->saddr, iph->daddr, 0, 0);
err = afinfo->route(net, dst, &flow, false);
[..]
if (err == 0 && dst->dev == par->in) -> OK
```

- module options: loose mode/valid mark/accept local
- also works with ipv6 pretty much same way
- BUT...

Problems, Part 2

- it is not the same as current code (ignores iif on route lookup)
- breaks with multipath routes:
  - When FIB has several possible routes, picks random result
  - no way to detect this condition from ”outside” of route core
  - i.e., `dst->dev != par->in` does not necessarily mean that rp-test has failed
  - IPv6 apparently does not have this problem (`strict` matching)

2nd Try

Try to do the same thing that `fib_validate_source` does

```
fib_lookup(net, &flowi4, &res));
[..]
```

- main problem: needs two `fib_lookup` calls (1st to find oif, 2nd to do actual rp test)

- OTOH: no problem with multiple paths

Other remaining issues & open questions

- builtin rpf does not care about ipsec-protected packets (`secpath_exists(skb) ? -> return`), should we?
- result caching – is it even worth the effort?
- should `loopback` be treated specially (e.g. ignored)?
- single module (`xt_rpf`) or seperate modules for ipv4/ipv6?
- move ttl/MTU/gateway tests after `FORWARD` hooks?

Summary

- running code for both ipv4/ipv6, tried different implementations (`fib_lookup`, `afinfo->route`)
- several problems remain
  - spoofed packet w. expired TTL/hoplimit causes icmp message; unless putting rpf check into `PRE_ROUTING`
  - . . . `PRE_ROUTING` sucks – need `oif` for reverse input lookup
  - . . . unless doing things differently and using output route lookup (not the same as current `fib_validate_source`
  - `FORWARD` is also the chain where rpf test would make the most sense
- might make sense to focus on ipv6 rpf module first